

Accessify - Plugin para analisar e corrigir violações de acessibilidade na web usando LLM

Simone de Fátima Skura¹, Murilo Marçal de Araujo²,
Franciele Beal¹, Marlon Marcon¹

¹Programa de Pós-Graduação em Informática (PPGI-CP/DV)
Universidade Tecnológica Federal do Paraná (UTFPR)
Multicampi Cornélio Procópio e Dois Vizinhos – PR – Brazil

²Coordenação de Engenharia de Software
Universidade Tecnológica Federal do Paraná (UTFPR)
Dois Vizinhos – PR – Brazil

simoneskura@gmail.com, murilo.araujo@alunos.utfpr.edu.br

{fbeal,marlonmarcon}@utfpr.edu.br

Abstract. *Web accessibility is essential for including visually impaired individuals in digital education. Many educational websites still do not comply with WCAG guidelines. This work presents Accessify, a browser plugin designed to automatically fix accessibility issues. The tool uses LLMs to process HTML and adapt the DOM in real time. All changes are performed locally preserving the original server code. The solution was evaluated across ten educational websites. The WAVE tool was used for accessibility assessment. Results reduced critical issues from 10,0 to 0.50 per page. The AIM Score increased from 5,72 to 7.41. Despite minor inconsistencies, Accessify was effective in reducing accessibility barriers.*

Resumo. *A acessibilidade web é essencial para incluir pessoas com deficiência visual na educação digital. Muitos sites educacionais ainda não seguem as diretrizes WCAG. Este trabalho apresenta o Accessify, um plugin de navegador para corrigir problemas automaticamente. A ferramenta utiliza LLMs para processar o HTML e adaptar o DOM em tempo real. As alterações são feitas localmente preservando o código original no servidor. A solução foi avaliada em dez sites educacionais. A ferramenta WAVE foi usada para a avaliação de acessibilidade. Os resultados reduziram problemas críticos de 10,0 para 0,50 por página. O AIM Score aumentou de 5,72 para 7,41. Apesar de pequenas inconsistências, o Accessify foi eficaz na redução de barreiras de acessibilidade.*

1. Introdução

A OMS estima que 2,2 milhões de pessoas no mundo tem algum tipo de deficiência visual [WORLD HEALTH ORGANIZATION 2026]. No Brasil, segundo o Censo 2022 [IBGE 2025] do Instituto Brasileiro de Geografia e Estatística (IBGE), são estimadas 14,4 milhões de pessoas com deficiência, entre elas, 7,9 milhões tem deficiência visual. Esses números evidenciam a relevância de iniciativas voltadas à inclusão e à acessibilidade digital, especialmente em contextos educacionais, nos quais o acesso equitativo à informação é fundamental para a aprendizagem.

Com a expansão de ambientes virtuais de aprendizagem, plataformas educacionais e materiais didáticos disponíveis na web, estudantes com deficiência visual dependem cada vez mais da acessibilidade das interfaces digitais para acessar conteúdos, interagir com sistemas e participar de atividades acadêmicas. Embora existam recomendações consolidadas para o desenvolvimento de conteúdos acessíveis, como as diretrizes do *Web Content Accessibility Guidelines* (WCAG¹), muitos sites e sistemas educacionais ainda apresentam barreiras que dificultam ou impedem a navegação por pessoas cegas ou com baixa visão. Estudos indicam que a maioria dos sites, de modo geral, não atende aos requisitos básicos de acessibilidade, resultando em dificuldades na navegação e compreensão do conteúdo por esses usuários [Huang et al. 2024], [WebAIM 2024], [Ara et al. 2025].

Nesse contexto, soluções que possam adaptar ou corrigir problemas de acessibilidade em páginas web tornam-se relevantes para ampliar o acesso de estudantes com deficiência visual a conteúdos educacionais. Este trabalho apresenta um plugin de navegador chamado de Accessify. Ele foi desenvolvido para identificar e corrigir violações de acessibilidade de páginas web para estudantes cegos. O plugin realiza adaptações no código HTML das páginas diretamente no navegador do usuário, conforme as recomendações das diretrizes WCAG, sem modificar o código original das plataformas ou realizar processamento em servidores. Como principal contribuição, este artigo apresenta **o projeto e a implementação de um plugin capaz de identificar e corrigir automaticamente problemas comuns de acessibilidade em páginas web**. Espera-se que a solução contribua para reduzir barreiras de acessibilidade em ambientes digitais de aprendizagem e incentive o desenvolvimento de tecnologias educacionais mais inclusivas.

2. Acessibilidade na Web

A deficiência visual é caracterizada pela perda total ou parcial da capacidade de enxergar em ambos os olhos, de forma permanente, que não pode ser melhorada ou corrigida por tratamento clínico, cirúrgico ou utilização de lentes. Ela é classificada em dois diferentes graus: baixa visão, que pode ser corrigida parcialmente com o uso de lentes de aumento ou lupas, e cegueira, quando a utilização de óculos ou lentes não é suficiente e as pessoas dependem de leitores de tela e outras tecnologias assistivas para acessar a internet [PARANÁ 2025].

Conforme o artigo 3º da Lei Brasileira de Inclusão, "tecnologia assistiva são produtos, equipamentos, dispositivos, recursos, metodologias, estratégias, práticas e serviços que objetivem promover a funcionalidade, relacionada à atividade e à participação da pessoa com deficiência ou com mobilidade reduzida, visando à sua autonomia, independência, qualidade de vida e inclusão social"[BRASIL 2015]. Dentre estas tecnologias, destacam-se os leitores de tela que são softwares que interpretam textos, botões e outros elementos da interface, transmitindo essas informações através da síntese de voz ou dispositivos braille. Os leitores de tela mais utilizados são NVDA², JAWS³, VOICEOVER⁴ e TALKBACK⁵.

¹WCAG - <https://www.w3c.br/traducoes/wcag/wcag22-pt-BR/>

²NV ACCESS - <https://www.nvaccess.org>

³JAWS Screen Reader - <https://www.freedomscientific.com>

⁴VoiceOver - <https://www.apple.com/accessibility/vision/>

⁵TalkBack - <https://support.google.com/accessibility/android/answer/6283677>

A acessibilidade web significa tornar sites, sistemas, aplicativos e conteúdos digitais acessíveis a todas as pessoas, sem levar em consideração as limitações físicas, sensoriais, cognitivas ou tecnológicas. Isso significa garantir que indivíduos com deficiência visual, física, motora, intelectual ou auditiva possam acessar e interagir com os conteúdos de forma independente e sem nenhuma restrição [Ribeiro and Lima 2020].

A WCAG é um conjunto de diretrizes e recomendações reconhecidas no mundo todo, elaborada pelo *World Wide Web Consortium* (W3C⁶) por meio da *Web Accessibility Initiative* (WAI⁷). O principal objetivo das diretrizes é orientar desenvolvedores, designers, criadores de conteúdo e outras partes interessadas na criação de websites e aplicações web que possam ser utilizados por todas as pessoas, independentemente de suas habilidades ou limitações. Sua base é formada por diretrizes que fornecem o que deve ser feito para tornar o conteúdo mais acessível. Além disso, apresenta três níveis de conformidade (A - mínimo exigido, AA - aceitável e AAA - ideal) para avaliar o quanto bem um conteúdo web atende as diretrizes [W3C BRASIL 2023].

O não cumprimento dessas diretrizes ocasiona problemas de acessibilidade na web que impedem ou dificultam que pessoas com deficiência visual utilizem sites e aplicações de forma independente. Com base na WCAG, os principais problemas são os seguintes: imagem sem texto alternativo, falta de estrutura semântica no HTML, navegação incompatível com o teclado, conteúdos dinâmicos não acessíveis, formulários mal rotulados, elementos interativos não padronizados, problemas de contraste, falta de descrição em gráficos ou mapas, impossibilidade de identificar o foco, leitura fora de ordem semântica, links e botões com textos genéricos, elementos visuais baseados somente em cor ou imagem, mudança de contexto sem aviso, textos gerados dinamicamente com Javascript sem notificar o leitor de tela, falta ou uso incorreto de Landmarks (*main*, *navi*, *header*), falta de acessibilidade nos leitores de CAPTCHA e tabelas sem cabeçalho ou escopo definidos [W3C BRASIL 2023].

Existem ferramentas automatizadas que auxiliam desenvolvedores na avaliação de acessibilidade web. Uma delas é a WAVE⁸ desenvolvida pela WebAIM (*Web Accessibility in Mind*), uma organização sem fins lucrativos vinculada da Universidade de Utah (*Utah State University*). Ela pode ser usada para identificar falhas técnicas e estruturais que impedem o acesso de pessoas com deficiência visual. Suas verificações estão conforme as diretrizes WCAG (versões 2.1 e 2.2 - níveis A e AA).

O foco da WAVE está na avaliação visual contextualizada, injetando ícones e indicadores diretamente na interface do usuário para facilitar a identificação de barreiras de interação. Ela pode ser usada como uma extensão do navegador. Para fazer a análise de um site, basta acioná-la. A ferramenta processa o conteúdo após a renderização do JavaScript e CSS, analisando o DOM (Modelo de Objeto de Documento) final. Indicadores visuais das falhas são injetados diretamente no DOM renderizado, possibilitando estabelecer uma conexão imediata entre a falha técnica e o elemento de interface que a originou. Essa visibilidade é relevante para a compreensão da arquitetura de informação, pois, ao expor a estrutura semântica de forma gráfica, a WAVE ensina o desenvolvedor a "enxergar" a página através da lógica de uma tecnologia assistiva. Portanto, a ferramenta não

⁶W3C - <https://www.w3.org/>

⁷WAI - <https://www.w3.org/WAI/>

⁸WAVE Web Accessibility Evaluation Tools - <https://wave.webaim.org/>

atua apenas como um validador de conformidade com a WCAG, mas como um recurso instrucional que promove a conscientização sobre o design inclusivo durante o processo de correção. A Figura 1 mostra a análise de acessibilidade com WAVE do site Gov.br. A esquerda está o painel da WAVE com os detalhes da avaliação e, ao centro, a página com os indicadores visuais injetados.

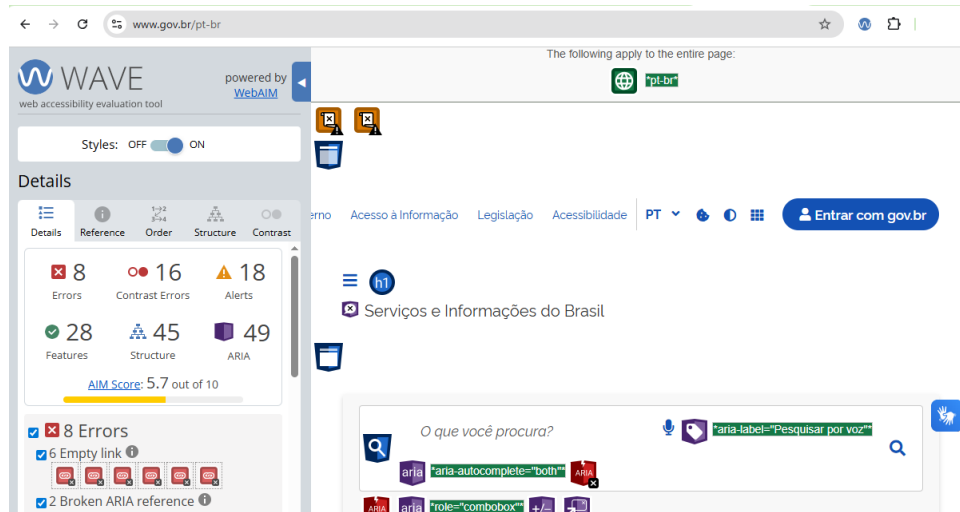


Figura 1. Demonstração do uso do WAVE no site GOV.BR.

Os resultados de uma avaliação WAVE são apresentados em seis categorias principais:

- **Errors (Vermelho):** Problemas críticos que violam a WCAG (ex: ausência de texto alternativo);
- **Contrast Errors (Vermelho):** problemas de contraste entre texto e fundo;
- **Alerts (Amarelo):** Elementos que podem causar problemas e exigem revisão humana;
- **Features (Verde):** Implementações de acessibilidade bem-sucedidas;
- **Structural Elements (Azul):** Identificação de marcos (landmarks) e hierarquia de títulos;
- **ARIA (Roxo):** Presença e exatidão de atributos de acessibilidade dinâmica.

A WAVE utiliza uma métrica de impacto, O AIM Score (*Accessibility Impact*), que varia de 1 a 10. Ela é calculada com base no número de erros detectados, densidade de erros (número de erros detectáveis por elementos da página) e número de alertas (possíveis ou prováveis problemas de acessibilidade). O cálculo é gerado automaticamente por meio de uma média ponderada composta por 60% da contagem total de erros, 30% da densidade de erros e 10% do número de alertas. O resultado é alinhado com os dados do WebAIM Million (análise anual de 1 milhão de sites [WebAIM 2026]). Um score de 5.0 indica que a página está na média exata da web mundial. Scores próximos de 10 indicam que a página está entre as mais acessíveis globalmente.

Neste trabalho, a WAVE será usada para fazer a avaliação de um conjunto de sites antes e após aplicar o plugin proposto. A ideia é verificar se a aplicação do plugin melhora o *score* de acessibilidade dos sites analisados.

3. Trabalhos Relacionados

Esta seção apresenta os trabalhos que deram ideias e serviram de inspiração para este trabalho. O primeiro deles, o trabalho de [Huang et al. 2024] apresenta uma abordagem para corrigir automaticamente violações de acessibilidade web, utilizando Modelo de Linguagem de Grande Porte (LLM) para modificar o DOM da página web em tempo real, usando o GPT-3.5 turbo para gerar código HTML para corrigir os erros de acessibilidade comuns. No trabalho de [Lee et al. 2020] é apresentado o iTOC, uma extensão de navegador que detecta e extrai o TOC (*Table of Contents*, ou Tabela de Conteúdos) do DOM de forma automática e o disponibiliza para usuários de leitores de tela via interface personalizada que pode ser acessada com tecla de atalho, onde, é possível utilizar as setas para navegar pelo conteúdo. O estudo de [Tiwary and Mahapatra 2022] criou uma solução para facilitar o acesso a imagens de um site abordando a geração automática de texto alternativo de imagens usando inteligência artificial. O trabalho realizado por [M S et al. 2024] desenvolveu uma extensão que injeta um código Javascript na estrutura da página, coleta as imagens, gera uma descrição e substitui o atributo *alt-text* da imagem pela descrição gerada. Já o trabalho de [Fathallah et al. 2025] visa detectar e corrigir as violações de acessibilidade por meio do código HTML. Eles propuseram um método que utiliza ferramentas de teste de acessibilidade e LLMs para detectar e corrigir as violações das diretrizes de acessibilidade web.

Ao observar a literatura, a maioria das ferramentas foca em aspectos específicos da acessibilidade (como geração de *alt-text*) sem abordar correções abrangentes de estrutura semântica. Além disso, muitas das soluções requerem processamento em servidores externos, levantando questões de privacidade e latência. A solução desenvolvida neste trabalho diferencia-se por oferecer correções abrangentes de acessibilidade por meio de uma extensão de navegador que processa o HTML usando LLMs. A Tabela 1 apresenta um comparativo entre os trabalhos relacionados e o plugin proposto, mostrando as limitações desses trabalhos e o diferencial do Accessify.

4. Ferramenta Desenvolvida (Accessify)

O Accessify é uma extensão de navegador baseada na arquitetura Manifest V3, projetada para identificar e mitigar dinamicamente violações de acessibilidade em aplicações web. O sistema atua diretamente na manipulação do *Document Object Model* (DOM) em tempo de execução (*runtime*), assegurando conformidade com as diretrizes WCAG 2.1 (níveis A e AA), com foco na otimização da experiência de usuários dependentes de tecnologias assistivas. O repositório com todo o código-fonte da aplicação está disponível em <https://github.com/ICDI/Accessify>, juntamente com informações de como instalar o plugin, como usar, informações de configurações, da arquitetura, privacidade e segurança, licença, entre outros.

4.1. Arquitetura e Componentes do Sistema

A arquitetura do Accessify foi concebida sob um paradigma híbrido que combina avaliação heurística local com processamento semântico baseado em Modelos de Linguagem de Grande Porte (LLMs). Essa abordagem minimiza a latência de rede e restringe o envio de dados a servidores externos, reforçando a privacidade e a segurança da informação. O sistema é estruturado em três componentes principais:

Tabela 1. Comparativo entre Trabalhos Relacionados e o Accessify

Trabalho	Abordagem	Limitações	Diferencial do Accessify
[Huang et al. 2024]	Correção automática de violações modificando o DOM em tempo real com GPT-3.5 turbo.	Foco em correções comuns, mas muitas soluções na área dependem de processamento em servidores externos, gerando latência.	Combina avaliação local determinística (reduzindo chamadas externas) com a flexibilidade de múltiplos modelos LLM via OpenRouter.
iTOC [Lee et al. 2020]	Extensão que extrai a Tabela de Conteúdos (TOC) para facilitar a navegação via leitor de tela.	Abordagem restrita a um aspecto específico da acessibilidade (navegação por índices).	Atua de forma abrangente na estrutura semântica (hierarquia, ARIA, contraste, imagens), não se limitando a índices.
[Tiwary and Mahapatra 2022]	Geração automática de texto alternativo para imagens usando Inteligência Artificial.	Foca exclusivamente no tratamento de imagens, ignorando falhas estruturais do HTML.	Além de gerar <i>alt text</i> , reestrutura a semântica da página inteira.
Websight [WebAIM 2024]	Injeção de JavaScript para coletar imagens e substituir o atributo <i>alt-text</i> por descrições geradas por IA.	Restrito ao aspecto visual (imagens), sem abordar correções abrangentes na estrutura do DOM.	Realiza a modificação local do DOM de forma ampla, englobando também textos, links genéricos e botões não rotulados.
Accessguru [Fathallah et al. 2025]	Utiliza ferramentas de teste em conjunto com LLMs para detectar e corrigir violações no código HTML.	O texto destaca que muitas soluções requerem processamento em servidores de terceiros, levantando questões de privacidade.	O <i>Content Script</i> faz a detecção inicial no próprio navegador e realiza as adaptações localmente.

- **Popup (Interface de Controle):** Desenvolvido em HTML e JavaScript (*popup.js*), atua como a interface de usuário (UI) e orquestrador de estado, gerenciando o ciclo de vida do *job* de análise e a renderização assíncrona dos resultados.
- **Content Script (content.js):** Injetado no contexto de execução da página web, é responsável pela inspeção do DOM, extração de metadados e aplicação das mutações corretivas no *client-side*, preservando a integridade do código-fonte original hospedado no servidor.
- **Background Service Worker (background.js):** Opera de forma assíncrona e isolada no *background*, gerenciando a persistência de estado via *Chrome Storage API* e mediando a comunicação HTTP *cross-origin* com a API de IA, evitando bloqueios na *main thread* da página.

4.2. Fluxo de Processamento e Integração com LLM

O diferencial técnico do Accessify reside na sua capacidade de transcender validações baseadas puramente em regras (expressões regulares e seletores estáticos), empregando LLMs para inferir o contexto semântico dos elementos da interface. O fluxo de dados da aplicação ocorre em quatro estágios orquestrados, conforme ilustrado na Figura 2.

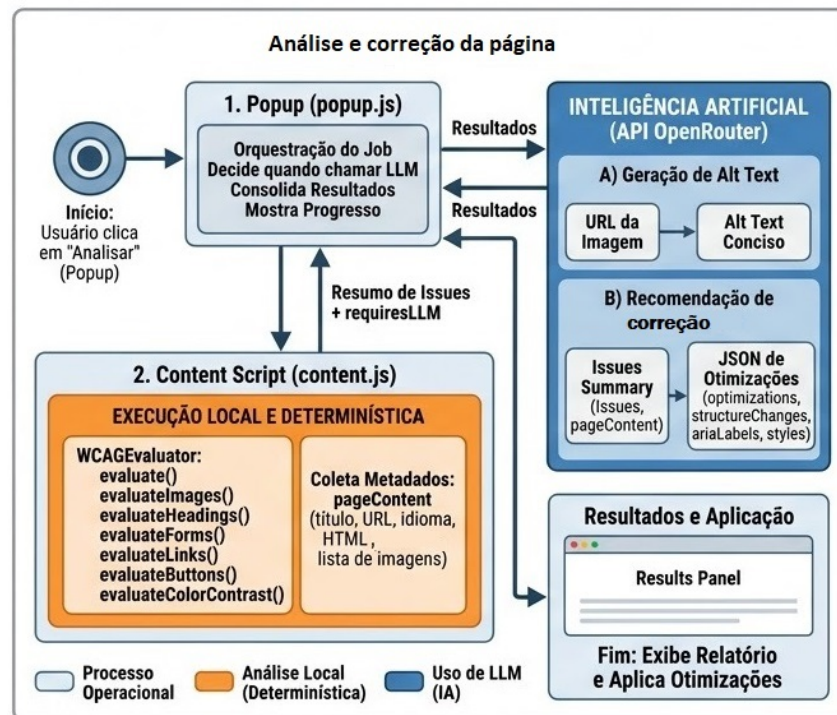


Figura 2. Arquitetura e Fluxo do Processamento

1. Orquestração e Despacho de Eventos (*popup.js*) O fluxo de execução é engatilhado por uma interação do usuário na interface da extensão. O *popup.js* assume a responsabilidade de orquestrar o *pipeline* de análise, despachando mensagens inter-processos (*Message Passing*) para o *Content Script* e gerenciando o estado da interface (etapas de progresso, resultados e consolidação de *logs*).

2. Avaliação Determinística e Extração de Contexto (*content.js*) Através de *listeners* de mensagens, o *Content Script* inicia a varredura local no DOM. Esta fase compreende duas operações executadas pela classe *WCAGEvaluator*:

- **Avaliação Heurística:** Executa algoritmos de validação estrutural no DOM (métodos como *evaluateImages()*, *evaluateHeadings()*, *evaluateForms()*, *evaluateColorContrast()*), identificando anomalias como ausência de atributos *aria-labels*, hierarquia de cabeçalhos inconsistente e contraste de cores abaixo do limiar exigido (4.5:1).
- **Geração de *Payload* (*pageContent*):** Serializa o estado atual da página, construindo um objeto contendo metadados essenciais (título, URL, idioma), um subconjunto do código HTML e uma matriz de referências de imagens.

Os resultados são compilados em um objeto *Issues Summary*, acompanhado de uma *flag* booleana (*requiresLLM*), indicando a necessidade de delegação de processamento cognitivo para a IA.

3. Processamento Semântico via IA (*background.js* e API OpenRouter) Quando intervenções contextuais são requisitadas, o *Service Worker* intercepta o *payload* e realiza requisições HTTP RESTful para o *endpoint* da API OpenRouter⁹. A arquitetura

⁹OpenRouter - <https://openrouter.ai/>

divide o processamento em dois fluxos paralelos:

- **A) Inferência de Visão Computacional (*Alt Text*):** URLs de imagens desprovidas de descrição são submetidas a modelos multimodais. A IA processa o conteúdo visual e retorna *strings* textuais descritivas, otimizadas para consumo por *Screen Readers*.
- **B) Refatoração Estrutural:** O *Issues Summary* e o *pageContent* compõem o *prompt* submetido ao LLM. O modelo atua como um analisador estático avançado, retornando um objeto estruturado em JSON contendo instruções de mutação do DOM, categorizadas em alterações estruturais (*structureChanges*), injeção de atributos de acessibilidade (*ariaLabels*) e modificações de folhas de estilo (*styles*).

4. Mutação Dinâmica do DOM e *Feedback* No estágio final, o orquestrador consolida os *outputs* das análises determinística e probabilística (IA), renderizando um *dashboard* de métricas para o usuário. Mediante a confirmação da ação de correção, a classe *AccessibilityAnalyzer* no *Content Script* aplica as mutações diretamente na árvore do DOM. A injeção de novos nós HTML, atualização de atributos ARIA e refatoração de hierarquia de títulos ocorrem de forma não-bloqueante, refletindo as otimizações na camada de apresentação em tempo real e aprimorando instantaneamente a navegabilidade. A Figura 3 apresenta uma representação gráfica de como o plugin atua para corrigir inconsistências nas páginas web.

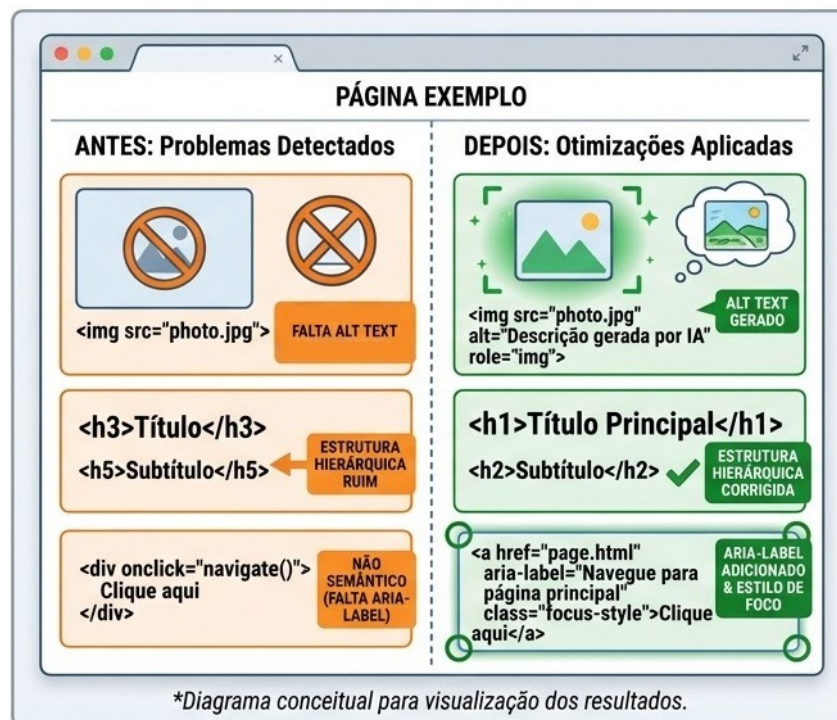


Figura 3. Exemplo de "Antes e Depois" da Aplicação das Otimizações

5. Avaliação e Resultados

Para verificar se a aplicação do Accessify corrige e melhora a acessibilidade de sites, foram selecionados 10 sites de conteúdo educacional, sendo a maioria deles utilizados por estudantes de graduação e pós-graduação da área de computação, são eles: Congresso

da Sociedade Brasileira de Computação (CSBC 2026), CAPES, Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais (IHC 2025), Congresso Brasileiro de Informática na Educação (CBIE 2025), Brazilian Conference on Intelligent Systems (BRACIS 2026), SBC OpenLib, Instituto Benjamin Constant (IBC), Escola Virtual Gov, Congresso Brasileiro de Software (CBSOFT 2026) e EduComp 2026. Foram analisadas as páginas iniciais de cada um deles. Inicialmente, foi usado a ferramenta WAVE que mostra os problemas atuais de acessibilidade (*Errors*, *Contrast Errors*, *Alerts*, *Features*, *ARIA*) e fornece o *AIM Score*, na página inicial de cada site. Em seguida, o Accessify foi aplicado na página inicial de cada um deles.

A Tabela 2 apresenta uma comparação antes (WAVE) e depois (Accessify) para diferentes métricas de acessibilidade nos 10 sites. A análise geral indica que o plugin teve impacto positivo na maioria dos casos, mas com variações importantes entre os sites e métricas.

Tabela 2. Análise de Acessibilidade (WAVE) - Antes e Depois com Médias

Site	Fase	Errors	Contrast Errors	Alerts	Features	Structure	ARIA	AIM Score
CSBC 2026	Antes	8	9	8	5	33	52	6.5
	Depois	0	8	4	5	33	46	8.4
CAPES	Antes	31	0	35	140	109	368	6.1
	Depois	0	0	56	140	109	497	9.3
IHC 2025	Antes	15	26	19	12	46	33	4.2
	Depois	0	49	15	12	46	40	3.3
CBIE 2025	Antes	19	37	7	30	58	106	3.8
	Depois	3	22	4	30	58	105	5.7
BRACIS 2026	Antes	1	11	8	13	19	23	6.2
	Depois	0	7	8	13	19	14	7.4
SBC Open Lib	Antes	0	11	0	5	14	8	5.8
	Depois	0	3	0	5	14	8	8.5
IBC	Antes	10	9	23	109	129	252	7.1
	Depois	2	11	22	109	129	360	7.8
Escola Virtual Gov	Antes	0	10	19	41	147	64	7.2
	Depois	0	5	17	41	147	13	8.1
CBSOFT 2026	Antes	11	7	41	10	36	43	5.5
	Depois	0	8	21	10	35	22	7.5
EduComp 2026	Antes	5	27	4	23	23	48	4.8
	Depois	0	7	3	25	23	44	8.1
MÉDIA GERAL*	Antes	10,0	14,7	16,4	38,8	61,4	99,7	5,72
	Depois	0,50	12,0	15,0	39,0	61,3	114,9	7,41

Em relação a métrica *Errors*, houve redução significativa de erros em vários sites: CSBC 2026, CAPES, IHC 2025, BRACIS 2026, CBSOFT 2026 e EduComp 2026, eliminação total; CBIE 2025 de 19 para 3; e, IBC de 10 para 2. Em alguns casos, os erros já inexistentes, por exemplo, SBC OpenLib e Escola Virtual Gov. Desta forma, pode-se dizer que o Accessify é eficaz na correção de erros básicos de acessibilidade.

Sobre a métrica *Contrast Errors*, os resultados mostram que o Accessify pode introduzir ou não corrigir adequadamente problemas de contraste, o que é problemático para acessibilidade visual. A métrica *Alerts* também não apresentou resultados significativos, reduzindo em alguns casos (CSBC 2026, IHC 2025, CBIE 2025, IBC, Escola Virtual Gov, CBSOFT 2026 e EduComp 2026) e aumentando em outro (CAPES). Porém, deve-se considerar que alertas dependem de contexto e nem sempre são resolvidos automaticamente, sendo necessária a correção manual por humanos.

As métricas *Features* e *Structure* apresentaram pouca alteração na maioria dos si-

tes, em alguns casos, valores permaneceram idênticos (antes/depois), ou seja, o Accessify não alterou significativamente a estrutura semântica ou *features* detectadas pelo WAVE.

A métrica *ARIA*, apresentou redução em alguns sites analisados. O site CAPES passou de 368 para 497 elementos, resultando em um aumento, o IHC 2025 subiu de 33 para 40 e o IBC de 252 para 360. Os demais sites apresentaram redução, com exceção do SBC Open Lib que se manteve. Assim, percebe-se que o plugin pode inserir ou modificar atributos *ARIA*, mas nem sempre de forma consistente.

O *AIM Score* apresentou melhora significativa em 9 sites: CSBC 2026 de 6.5 para 8.4, CAPES de 6.1 para 9.3, CBIE 2025 de 3.8 para 5.7, BRACIS 2026 de 6.2 para 7.4, SBC OpenLib de 5.8 para 8.5, IBC de 7.1 para 7.8, Escola Virtual Gov de 7.2 para 8.1, CBSOFT 2026 de 5.5 para 7.5 e EducComp 2026 de 4.8 para 8.1. Piora no IHC 2025 que passou de 4.2 para 3.3. Percebe-se que o Accessify tende a melhorar a acessibilidade global, mas ainda não de forma uniforme. Entretanto, o Accessify demonstrou ser eficaz na redução de erros básicos e na melhoria do score geral de acessibilidade.

A avaliação dos 10 sites processados demonstra que o Accessify teve um impacto predominantemente positivo em reduzir barreiras críticas. A média de violações graves de acessibilidade (*Errors*) caiu de 10,0 para 0,50 por página. onsequentemente, a ferramenta promoveu um aumento médio no *AIM Score* global, que passou de 5,72 para 7,41. No entanto, a análise revela que atributos dependentes de validação contextual podem sofrer variações indesejáveis. Houve um aumento médio nas marcações *ARIA* (de 99,7 para 114,9) evidenciando que, embora a injeção automatizada de código com LLMs resolva deficiências primárias com eficácia, ela ainda pode introduzir inconsistências em regras estruturais mais complexas.

6. Conclusão

Este trabalho apresentou o Accessify, um plugin de navegador desenvolvido com o propósito de ampliar o acesso de estudantes com deficiência visual a conteúdos educacionais na web. Através de uma abordagem que combina verificações heurísticas locais com o uso de LLMs, a ferramenta demonstrou-se capaz de identificar e aplicar correções estruturais diretamente no DOM do usuário, alinhadas às diretrizes WCAG, sem a necessidade de alterar os códigos originais das plataformas.

A avaliação da ferramenta em portais acadêmicos e educacionais revelou resultados promissores, destacando-se a drástica redução das infrações mais críticas (*Errors*), comprovando a eficácia do plugin na correção de falhas primárias de acessibilidade. Contudo, os testes também evidenciaram limitações inerentes às abordagens baseadas em inteligência artificial generativa. A manipulação de elementos dependentes de percepção visual (como contrastes) ou de semântica de estado (como atributos *ARIA*) resultou, em determinados casos, na introdução de novas inconsistências, reforçando que alertas e contextos específicos ainda demandam revisão humana.

Apesar das ressalvas e desafios técnicos remanescentes, espera-se que abordagens orquestradas localmente, como a proposta pelo Accessify, contribuam ativamente para a redução de barreiras digitais, promovendo ambientes de aprendizagem online mais inclusivos e autônomos.

7. Trabalhos Futuros

Como desdobramentos desta pesquisa, propõe-se os seguintes trabalhos futuros:

- **Avaliação com Usuários Reais:** Realizar testes de usabilidade práticos com estudantes com deficiência visual (cegueira e baixa visão). A validação empírica com leitores de tela, como NVDA, JAWS, VoiceOver ou TalkBack, é essencial para verificar se as alterações no DOM, a nova estrutura de cabeçalhos e as descrições de imagens geradas pela IA são semanticamente compreensíveis e úteis na navegação real. Além disso, é necessária a avaliação de uso em formulários complexos e conteúdos dinâmicos.
- **Refinamento das Correções Geradas por IA:** Ajustar a lógica de orquestração e os *prompts* enviados aos modelos de linguagem (LLMs) para aprimorar a precisão das alterações de código. O objetivo é mitigar a introdução de novos problemas de contraste e a inserção inconsistente ou excessiva de atributos ARIA, efeitos colaterais observados durante a etapa de avaliação automatizada. Avaliar a realização de processamento ideterminístico desacoplado do LLM para mitigar tais problemas.
- **Investigação de Compatibilidade:** Analisar a arquitetura de sites que apresentaram falhas de processamento durante os testes, como o Currículo Lattes e o portal institucional da SBC. Isso permitirá aprimorar a robustez do *Content Script* do Accessify para lidar com bloqueios de segurança, estruturas complexas e aplicações de página única (SPAs).
- **Tratamento Híbrido de Alertas:** Desenvolver recursos de interação no painel da extensão que auxiliem o usuário na resolução da métrica de *Alerts*. Visto que esses alertas dependem fortemente de contexto e não apresentaram redução significativa apenas com a automação, uma abordagem colaborativa entre o usuário e a IA pode elevar os níveis de conformidade com a WCAG.
- **Viabilidade Financeira:** Realizar uma análise sobre os custos operacionais da ferramenta, que trate sobre o modelo financeiro das requisições via API do Open-Router e a viabilidade econômica do plugin para uso diário extensivo.

Uso de ferramentas de IA

Durante a elaboração deste trabalho, ferramentas de Inteligência Artificial foram utilizadas em conformidade com as diretrizes do evento, com as seguintes finalidades:

- **Gemini:** Utilizado para a geração das Figuras 2a e 2b.
- **GPT 5.4 (integrado ao GitHub Copilot):** Atuou como auxiliar na geração e sugestão de trechos de código para a aplicação desenvolvida.
- **Grammarly, Gemini e ChatGPT:** Empregados para a revisão ortográfica, gramatical e aprimoramento textual do artigo.

Referências

- Ara, J., Sik-Lanyi, C., Kelemen, A., and Guzsvinecz, T. (2025). An inclusive framework for automated web content accessibility evaluation. *Universal Access in the Information Society*, 24(2):1581–1607.
- BRASIL (2015). Lei nº 13.146, de 6 de julho de 2015. lei brasileira de inclusão da pessoa com deficiência (estatuto da pessoa com deficiência). Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/l13146.htm. Acesso em: 1 jun. 2025.

- Fathallah, N., Hernández, D., and Staab, S. (2025). Accessguru: Leveraging llms to detect and correct web accessibility violations in html code. Disponível em: <https://arxiv.org/html/2507.19549v1>. Acesso em: 11 mar. 2026.
- Huang, C., Ma, A., Vyasamudri, S., Puype, E., Kamal, S., Garcia, J. B., Cheema, S., and Lutz, M. (2024). Access: Prompt engineering for automated web accessibility violation corrections. Disponível em: <https://arxiv.org/abs/2401.16450>. Acesso em: 11 mar. 2026.
- IBGE (2025). Censo 2022: Brasil tem 14,4 milhões de pessoas com deficiência. Disponível em: <https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/43463-censo-2022-brasil-tem-14-4-milhoes-de-pessoas-com-deficiencia>. Agência de Notícias do IBGE. Acesso em: 10 mar. 2026.
- Lee, H.-N., Uddin, S., and Ashok, V. (2020). itoc: Enabling efficient non-visual interaction with long web documents. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3799–3806.
- M S, S. R., Joy, E., and J, L. S. (2024). Websight: An ai-based approach to enhance web accessibility for the visually impaired. In *2024 International Conference on Science Technology Engineering and Management (ICSTEM)*, pages 1–7.
- PARANÁ (2025). Deficiência visual. <https://www.desenvolvimentosocial.pr.gov.br/Pagina/Deficiencia-Visual>. Governo do Estado do Paraná. Disponível em: <https://www.desenvolvimentosocial.pr.gov.br/Pagina/Deficiencia-Visual>. Acesso em: 1 jun. 2025.
- Ribeiro, S. and Lima, D. (2020). Inclusão digital e acessibilidade na web. *Revista Inclusão*. Disponível em: <https://revista.ibict.br/inclusao/articloe/view/1678>. Acesso em: 1 jun. 2025.
- Tiwary, T. and Mahapatra, R. P. (2022). Web accessibility challenges for disabled and generation of alt text for images in websites using artificial intelligence. In *2022 3rd International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, pages 1–5.
- W3C BRASIL (2023). Diretrizes de acessibilidade para conteúdo web (wcag 2.2). Disponível em: <https://www.w3c.br/traducoes/wcag/wcag22-pt-BR/>. Acesso em: 2 jun. 2025.
- WebAIM (2024). The webaim million: An accessibility analysis of the top 1,000,000 home pages. Disponível em: <https://webaim.org/projects/million/>. Acesso em: 11 mar. 2026.
- WebAIM (2026). The webaim million - an annual accessibility analysis of the top 1,000,000 home pages.
- WORLD HEALTH ORGANIZATION (2026). Blindness and vision impairment. Disponível em: <https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment>. Acesso em: 10 mar. 2026.